IN THE CLAIMS:

1. *(currently amended)* A method of performing natural language generation, the method comprising the steps of:

selecting a reference grammar;

applying an input dependency tree to a tree choosing module ~~for~~, the tree choosing module using a stochastic tree model to select syntactic realizations for each node and forming a ~~in the~~ derivation tree;

producing a word lattice for the stochastically selected syntactic realizations in the derivation tree, the word lattice comprising all possible word sequences permitted by the input dependency tree structure, the chosen stochastically selected syntactic realizations, and the reference grammar; and

choosing a linear precedence output string of least cost from the produced word lattice.

2. *(currently amended)* The method as defined in claim 1 wherein an extended tree-adjoined grammar (XTAG) grammar is selected as the reference grammar.

3. *(original)* The method as defined in claim 1 wherein the Viterbi algorithm is used to chose the output string from the word lattice.

4. *(currently amended)* A natural language generator for translating an input dependency syntax tree into a natural language output, the generator comprising

a tree choosing module, responsive to the input dependency syntax tree, for stochastically selecting ~~syntactic realizations~~ tree-adjoining grammar trees for each node in the input dependency tree to create a semi-specified derivation tree, the tree choosing module including a tree module database for use in selection;

an unraveling module, responsive to the stochastically selected tree-adjoining grammar trees created by the tree choosing module and including a predetermined reference grammar database for creating from the syntactic realizations a lattice of all possible linearizations of said trees using the reference grammar of said database; and a linear precedence chooser module for selecting the most likely traversal through the lattice as the natural language output of the generator.

5. *(original)* The generator as defined in claim 4 wherein the linear precedence chooser module utilizes the Viterbi algorithm to select the most likely traversal path.

6. *(original)* The generator as defined in claim 4 wherein the unraveling module includes a reference grammar database.

7. *(original)* The generator as defined in claim 6 wherein the reference grammar database comprises an XTAG grammar database.